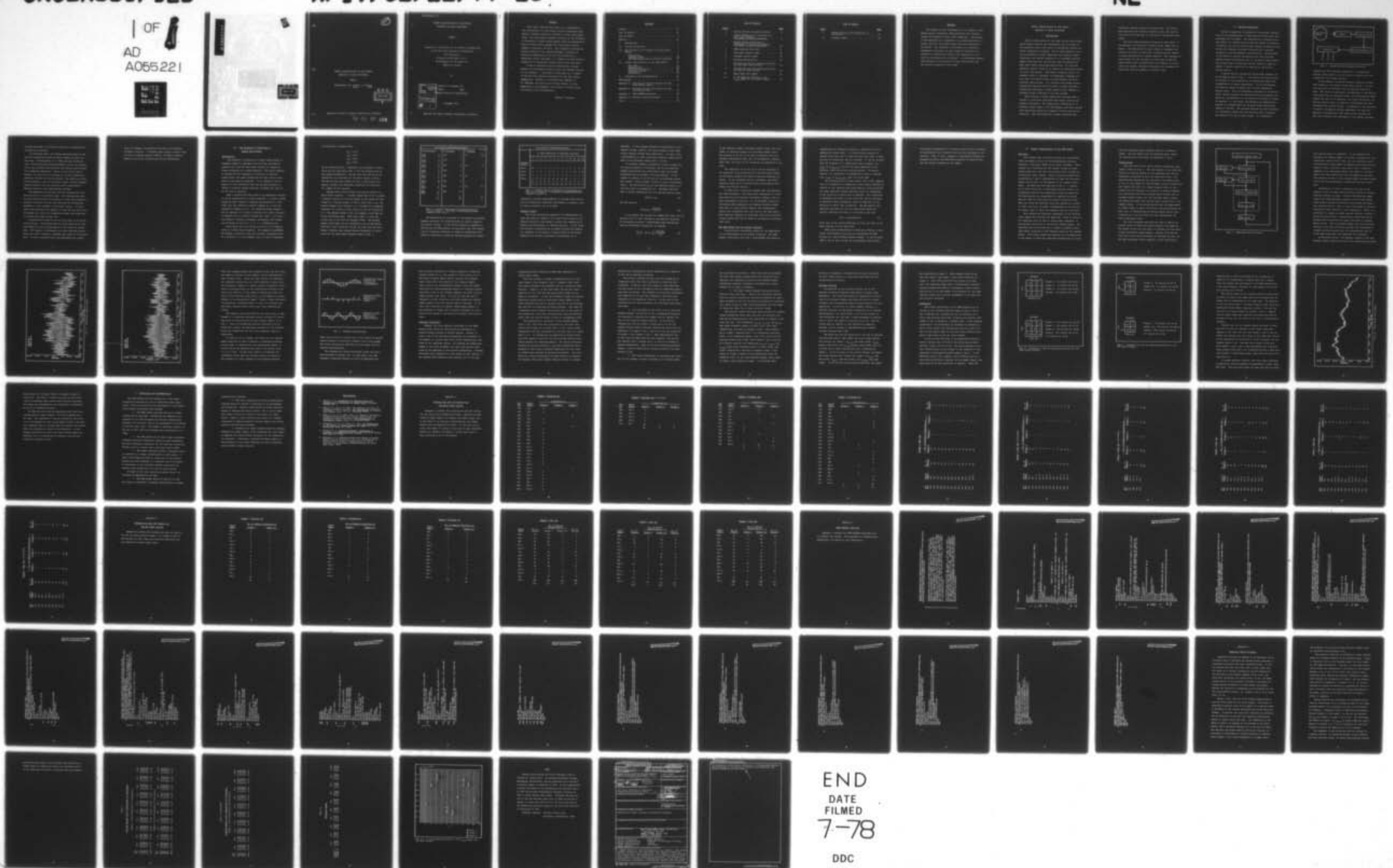
AD-A055 221 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 17/9
TARGET CLASSIFICATION BY TIME DOMAIN ANALYSIS OF RADAR SIGNATUR--ETC(U)
DEC 77 D G KULCHAK
UNCLASSIFIED AFIT/GE/EE/77-25 NL

1 OF 1
AD
A055221



END
DATE
FILMED
7-78
DDC

TARGET CLASSIFICATION BY TIME DOMAIN

ANALYSIS OF RADAR SIGNATURES

THESIS

AFIT/GE/EE/77-25 Delbert G. Kulchak
Major USAF

Approved for public release; distribution unlimited.

AFIT/GE/EE/77-25

TARGET CLASSIFICATION BY TIME DOMAIN ANALYSIS OF RADAR SIGNATURES

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Delbert G. Kulchak, B.S.
Major USAF
Graduate Electrical Engineering

December 1977

Preface

This report contains the result of an investigation into the ability of a time domain pattern recognition algorithm to classify targets by analysis of their radar signatures. The time domain procedure evaluated is the Frequency of Occurrence of Binary Words method, which was developed by Peter W. Becker while engaged with the General Electric Company in Syracuse, New York. The recognition problem was jointly proposed by Dr. Matthew Kabrisky, Professor of Electrical Engineering at the Air Force Institute of Technology (AFIT), and Major C. V. Stewart, an AFIT doctoral student who is preparing a dissertation in the same area.

I wish to express my sincere appreciation to Major Stewart for providing the data and for his tutelage throughout the problem. I also wish to thank Major R. A. Gagnon of the Air Force Avionics Laboratory for his many timely suggestions. Finally, I am particularly indebted to Dr. Kabrisky, who served as my thesis advisor, for his suggestions, encouragements, and infinite patience during the course of this investigation.

Delbert G. Kulchak

Contents

List of Figures

List of Tables

Abstract

The purpose of this investigation is to examine a time domain pattern recognition algorithm which will identify targets by analysis of their radar signatures. The essence of the algorithm is a conversion of an analog waveform to a binary sequence, from which binary words are selected as attributes. The selection of attributes is a heuristic, nonexhaustive process, and each target is eventually described in a statistical sense by the frequency of occurrence of the binary word attributes. A relationship between this Frequency of Occurrence of Binary Words method and the Fourier transform is also shown.

TARGET CLASSIFICATION BY TIME DOMAIN ANALYSIS OF RADAR SIGNATURES

I. Introduction

Recent investigations of the radar returns from illuminated targets indicate that information may be present in the signatures which could serve to distinguish between different categories of targets. One is often tempted to pursue the analysis of radar signatures via frequency domain techniques; the Fourier transform is an extremely powerful signal processing tool and has been used successfully in many radar signal processing applications. An alternate approach, however, would be to examine the signatures entirely in the time domain. Time domain techniques tend to be somewhat easier to implement in programming languages and also tend to be more suitable for minicomputer and microprocessor applications. If a suitable time domain pattern recognition algorithm could be found, it might represent a considerable advantage in these respects when compared to frequency domain solutions of the same problem.

Radar returns of three targets were recorded under a variety of conditions associated with target velocity and antenna orientation. The targets were illuminated at 16 different aspect angles, 0 to 360 degrees, with each successsive observation spaced 22.5 degrees from the previous observation. With few exceptions, target velocities were

relatively constant during the observations. The returns were digitized and recorded on magnetic tape, and used in this form as the raw data for the pattern recognition algorithm.

The time domain procedure evaluated in this report is the Frequency of Occurrence of Binary Words (FOBW) (Ref 1) method. The FOBW method is quite simple to implement and requires only that the data under consideration be capable of being represented as a binary sequence. This sequence is then scanned for the frequency of occurrence of specific binary words, with the expectation that members of a given class will exhibit different frequencies of occurrence for some words than will members of another class.

II. Pattern Recognition

Pattern recognition is becoming an increasingly valuable tool in the mechanization of tasks which previously have been performed only by humans. The applications of machines with the ability to classify and sort data are almost too numerous to mention, and include such diverse subjects as speech recognition and synthesis, waveform interpretation, character recognition, and photographic interpretation. Contributions to the growth of pattern recognition have come from many equally diverse disciplines, and, in the eyes of many experts, the ultimate goal of emulating certain classes of human response to sensory inputs may be realized in the not-too-distant future.

A typical pattern recognition system might resemble the block diagram shown in Fig. 1. The analog real-world energy of interest is first detected by an appropriate sensor and transmitted to a signal preprocessor. In the preprocessor, the detected energy is mapped into a finite dimensional pattern space. From the information available in the pattern space, certain features are extracted which the designer suspects are representative of the classes of patterns he wishes to separate. At this point, the designer has significantly reduced the dimensionality of the space which contains the energy of interest. The selected features are then presented to a classifier, which uses some decision rule to separate the features into two or more classes. It is generally

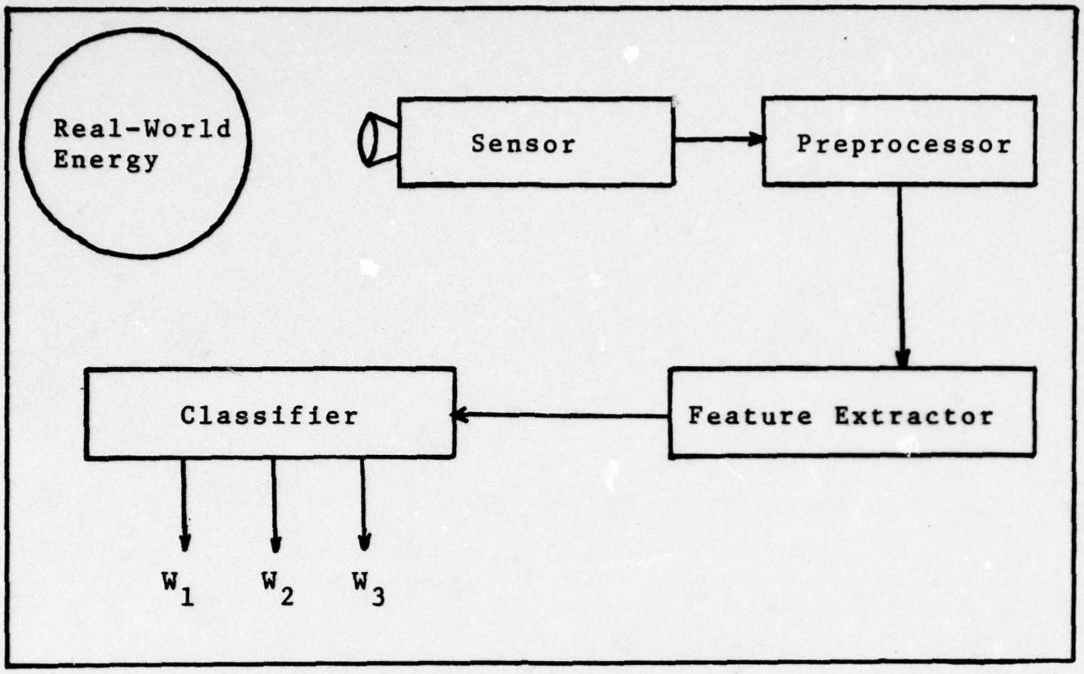


Fig. 1. Typical Pattern Recognition System.

assumed that the classifier operates in a deterministic fashion; there appear to be few practical applications for classifiers with random outputs.

In the design of an actual pattern recognition system, each element of the typical system shown in Fig. 1 is usually selected in accordance with the specific problem at hand. The sensor is selected by consideration of the physics associated with the observed energy, the preprocessing should enhance the data to be separated, and, most importantly, the feature extractor must be capable of determining what best represents the various classes. In addition to the selection of pattern recognition systems components, one must also generate a representative data base, select training and test sets, evaluate the performance of the system, and make

as many iterations in the entire process as is required for satisfactory operation.

As mentioned above, the feature selection phase of the pattern recognition process is nearly always the most crucial step. Unfortunately, it is often the most difficult, also. Poorly selected features generally add to the complexity of the classifier and decision rule and may make successful recognition impossible. There is very little theory, however, to guide one in a decision as to what constitutes effective features in a given problem. The search for effective attributes often is a heuristic affair, and the designer usually relies on his own intuition after considering the physical nature of the classification problem.

The separation of the data base into training and test sets can also be a nontrivial task. The training data must be sufficiently varied and inclusive to allow the recognition system to practice, yet the test data must be sufficiently different to give credence to the final results. It would serve little purpose, for example, to draw the test data from the design data since the recognition system could then have an artificially high success rate.

A performance evaluation is the final step in the recognition process. Some method must be chosen which would indicate whether or not the performance of the system is acceptable. The indices of performance are often specified beforehand, and, after a fashion, influence the choice of a decision rule. If one is concerned only with minimizing the overall

risk, for example, the decision rule may be an extension of Bayes' criterion. If minimal false alarm or reject rates are also of primary concern, however, the Neyman Pearson method or one's own criterion may be more appropriate.

III. The Frequency of Occurrence of Binary Words Method

Introduction

The Frequency of Occurrence of Binary Words method is elegantly simple to implement and, with few constraints, requires only that the data under analysis be capable of being represented as a binary sequence. The binary sequence is scanned for the frequency of occurrence of specific binary words, and these frequencies are then used as attributes in the pattern recognizer. It is assumed in the remainder of this discussion that the raw data consists of analog or discrete analog waveforms, although this need not be the case in practice.

When a waveform has been coded in an appropriate fashion, it can be represented as a binary sequence. A typical coding algorithm might consist of sampling the waveform at a constant rate and coding all positive samples as a 1 and all nonpositive samples as a 0. The resulting binary sequence may be regarded as a string of adjacent bits, where each pair of adjacent bits is called a digram (Ref 1:76). In a similar fashion, trigrams, tetragrams, pentagrams, and n-grams can be formed by sets of 3, 4, 5, ..., n adjacent bits.

Each n-gram can occur as any or all of its 2^n permutations in a given binary sequence. The sequence 11111000000, for example, contains 11 bits and 10 pairs of adjacent bits. If n is equal to 2, four digrams occur, or have a frequency

of occurrence, as shown below.

$$f_{11} = 4/10$$

$$f_{10} = 1/10$$

$$f_{01} = 0/10$$

$$f_{00} = 5/10$$

In the notation above, f represents the frequency of occurrence and the subscripts refer to the two adjacent bits in each digram permutation. One may note that the sum of the frequencies of occurrence for any given n-gram is unity. This may be seen more readily in Fig. 2, which shows the digram, trigram, and tetragram frequencies of occurrence in a sample 32 bit sequence.

Digrams which contain delays also may be present in a given binary sequence. A delayed digram (Ref 1:79) is simply a digram in which the two bits making up the digram are separated by a variable number of "don't care" bits, e.g., the delayed digram 1-m-0 consists of a 1 followed by m sampling intervals (or m minus 1 bits), which in turn is followed by a 0. The delayed digram 1-3-0, for example, could take any of the following forms: 1000, 1010, 1100, or 1110, and occurs once in the five bit sequence 10101. The delayed digram 1-m-0 might also be considered to be an (m+1)-gram with don't care states for all but the first and last bits. Figure 3 depicts some delayed digram frequencies of occurrence for the same binary sequence shown in Fig. 2.

11111100011111000000000000000011					
16 Tetragrams		**8 Trigrams**		**4 Digrams**	
1111	5	111	7	11	10
1110	2				
1101	0	110	2		
1100	2				
1011	0	101	0	10	2
1010	0				
1001	0	100	2		
1000	2				
0111	1	011	2	01	2
0110	0				
0101	0	010	0		
0100	0				
0011	2	001	2	00	17
0010	0				
0001	2	000	15		
0000	13				
	29		30		31

Fig. 2. n-gram Frequencies of Occurrence for the 32 Bit Sequence: 11111100011111000000000000000011 (Adapted from Ref 1).

The frequencies of occurrence of the n-grams and delayed digrams constitute the features which are used to separate the various classes. If the initial problem is amenable to solution by the FOBW method, one may expect that the frequencies of occurrence (referred to simply as frequencies hereafter) of particular n-grams and delayed digrams will remain

11111100011111000000000000000000011													
	m, The Separation in Sampling Intervals												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Delayed Digram													
1-m-1	10	7	5	4	3	3	4	5	5	4	3	2	1
1-m-0	2	4	6	7	8	8	7	6	6	7	8	9	10
0-m-1	2	4	5	5	5	4	3	2	2	2	2	2	2
0-m-0	17	15	13	12	11	11	11	11	10	9	8	7	6
	31	30	29	28	27	26	25	24	23	22	21	20	19

Fig. 3. Delayed Digram Frequencies of Occurrence for the 32 Bit Sequence: 11111100011111000000000000000000011 (Adapted from Ref 1).

essentially constant among members of the same class and yet will be sufficiently different from members of another class to allow a decision rule to be made.

Sequence Length

One of the constraints imposed by the FOBW method concerns the physical length of the coded binary sequence; the sequence should be long enough to permit the computation of binary word frequencies to a desired accuracy. If the original waveform is generated by an ergodic process and sampled at a harmonic of its period, a single period in the binary sequence will provide as accurate a calculation as is

possible. If the original waveform is generated by a stochastic process, however, the actual length of the coded binary sequence becomes more significant. In such a case, a determination of what constitutes adequate length may be made in the following fashion (Ref 1: 91-94).

If a binary sequence containing L bits is scanned for an n-gram frequency of occurrence, and if the intervals between observations are sufficiently long, the n-gram frequencies may be assumed to be uncorrelated. If the n-gram of interest is of the form ψ, the scanning process will reveal Γ total n-grams, of which γ are of the desired form ψ. The probability P_ψ that any observed n-gram is of the form ψ may be estimated by γ/Γ. Davenport and Root (Ref 4: 84-85) demonstrate that γ/Γ has the expected value

$$E[\gamma/\Gamma] = P_\psi \tag{1}$$

and the variance

$$\mathrm{Var}[\gamma/\Gamma] = \frac{P_\psi(1-P_\psi)}{\Gamma} \tag{2}$$

It is assumed that P_ψ does not change with time, and one may note that γ/Γ has a Bernoulli distribution regardless of the waveform process. Substituting these expressions into the Tchebycheff inequality, one obtains

$$P\left[\,\left|\frac{\gamma}{\Gamma} - P\psi\right| \geq \varepsilon\right] \leq \frac{P_\psi(1-P_\psi)}{\Gamma\varepsilon^2} \tag{3}$$

As the sequence length L increases without limit, the total number of observed n-grams also increases without limit. The variance in Eq (2) can be seen to approach zero as L becomes sufficiently large, and, by the Bernoulli theorem (Ref 4:85), the value of γ/Γ converges in probability to P_ψ, also.

The inequality in Eq (3) can be used to determine if the estimates of P_ψ associated with the particular length of the binary sequence are within a desired confidence interval. If the estimates are not acceptable, the binary sequence length may be increased until the estimates fall within the desired interval.

To this point, it has been assumed that the waveforms are time stationary. The binary word frequencies of occurrence may still be effective attributes, however, even if this requirement is not met. If the parameter values for the process which generated the original waveform change with time in a well defined fashion, the instantaneous binary word frequencies will also vary in the same manner. Each observed frequency of occurrence would then constitute a time average that may in itself be an effective attribute (Ref 1:94).

The FOBW Method and the Fourier Transform

A particularly interesting feature of the FOBW method is its relationship to the Fourier transform. One might suspect intuitively that such a relationship does exist by

considering the information present in symmetrical and unsymmetrical binary words. If a given binary sequence is scanned first from left to right and then from right to left, two sets of frequencies will be recorded. It can be verified that the frequency of a symmetrical word or digram, e.g., f_{1001} or $f_{0-m-1-m-0}$, will be the same regardless of the direction taken during the scanning process. In general, however, the frequencies of unsymmetrical words or digrams, e.g., f_{1101} or $f_{0-m-1-m-1}$, will not be the same.

One interpretation of these results (Ref 1:109) suggests that the frequencies of symmetrical words contain information similar to that obtained from power spectra, autocorrelation, and other amplitude related functions which are independent of the direction in which time is measured. The frequencies of unsymmetrical words, on the other hand, may be construed to represent phase information, which is sensitive to the direction in which time is measured. One may observe that amplitude and phase are the two terms which define the Fourier transform F(f) when it is written in the form

$$F(f) = A(f)\exp(j\theta(f)) \tag{4}$$

where A(f) is the Fourier amplitude of F(f), and $\theta(f)$ is the phase function of F(f) (Ref 6:78).

While this interpretation is admittedly liberal, a more formal relationship can be seen by considering the FOBW process as a zero-crossing analysis method. It can be shown (Refs 2 and 3) that, except for an arbitrary scale factor,

the Fourier coefficients of a function whose Fourier transform is band-limited can be derived from the zero-crossings of the function. This, in turn, suggests a relationship between the mechanism by which the FOBW method operates on waveform data and the Fourier transform of the signal.

IV. Target Classification by the FOBW Method

Data Base

The original data collection process was accomplished under government contract by the Environmental Research Institute of Michigan (ERIM) of Ann Arbor, Michigan. Three targets were illuminated as they passed through an established range gate, and the return spectra were recorded at 16 different aspect angles. Target velocities ranged from approximately 5 mph to approximately 15 mph, and the aspect angles varied from 0 to 360 degrees in 22.5 degree increments. The ERIM data was digitized by Maj C. V. Stewart, an Air Force Institute of Technology doctoral student, in conjunction with his own dissertation using the same data. The digitization was accomplished by sampling the analog data at 2 KHz and recording the results on magnetic tape. Both the analog and digitized data were labeled as a series of runs, with each run number corresponding to the particular aspect angle at which the original data was collected.

Both inphase and quadrature components of the received video signal were recorded and digitized. Since no particular advantage could be seen to using either component, it was decided arbitrarily to process the inphase signal. The digitized data was divided into a number of samples, where each sample consisted of 1024 adjacent points in the inphase signal. This sample length was selected for convenience due to the manner in which the tapes were prepared and to insure

that the resulting binary sequence would be of adequate length. The run numbers and corresponding aspect angles are tabulated for each target in Appendices A and B.

Preprocessing

The flow chart of the entire pattern recognition algorithm is shown in Fig. 4. The main program of the algorithm selects the desired samples of the digitized data and all subsequent processing is done via subroutine calls. The preprocessing subroutines are enclosed by the dashed lines in Fig. 4; each sample is time averaged, velocity normalized, and clipped and coded to a binary sequence which is then scanned for the frequency of occurrence of delayed digrams. Plot subroutines were also developed which display the unaveraged inphase signal (I), the averaged inphase signal, the quadrature signal (Q), and the phase (Arctan Q/I) of the complex signal. It was stated earlier that the FOBW method is rather simple to apply in practice, and preprocessing the data prior to implementing the delayed digram search algorithm proved to be more difficult than implementing the search algorithm itself.

The time averaging subroutine was used to remove the ground clutter in which the target signature was immersed. The ground clutter had the effect of forming a DC bias which varied considerably among samples. Digital filtering may have been a more effective method of removing the bias, but the time averaging routine appeared to work sufficiently

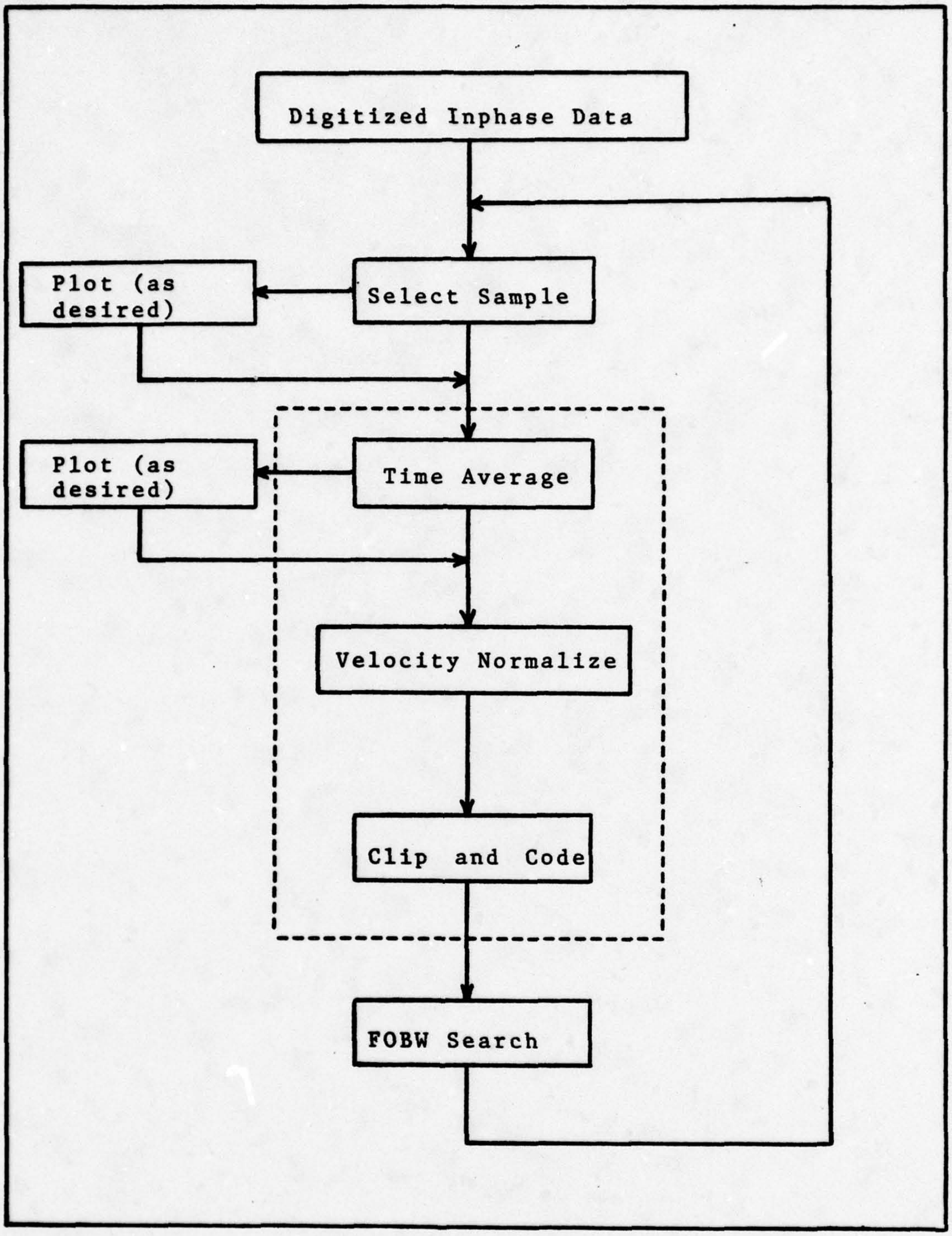


Fig. 4. FOBW Algorithm Flow Chart.

well and was easier to implement. It was observed that averaging the inphase signal in 64 point increments not only effectively removed the DC bias, but also produced a waveform which had a high degree of symmetry about the zero axis. Figures 5 and 6 illustrate this effect; Fig. 5 shows the asymmetry of an unaveraged sample from target 1, and Fig. 6 depicts the same sample after time averaging has occurred. This symmetry feature in the averaged signal was quite useful in the velocity normalization subroutine which was called next.

Intuitively, it would be desirable for each of the targets to have the same uniform velocity before any identification attempts are made. Velocity differentials among targets and among individual samples from the same target would produce radically different zero-crossing and binary word characteristics. This, in turn, would make the feature selection process difficult, if not impossible. Since it is unrealistic to impose a single constant velocity criteria in practical applications, a normalization subroutine was developed which normalizes all target velocities to approximately 18 mph. This is somewhat higher than the velocities observed in the data collection process, but normalizing to a higher velocity insures that all normalizations will be in the same "direction" and simplifies the algorithm.

As mentioned earlier, the symmetry feature of the time averaged inphase signal was used in the normalization process.

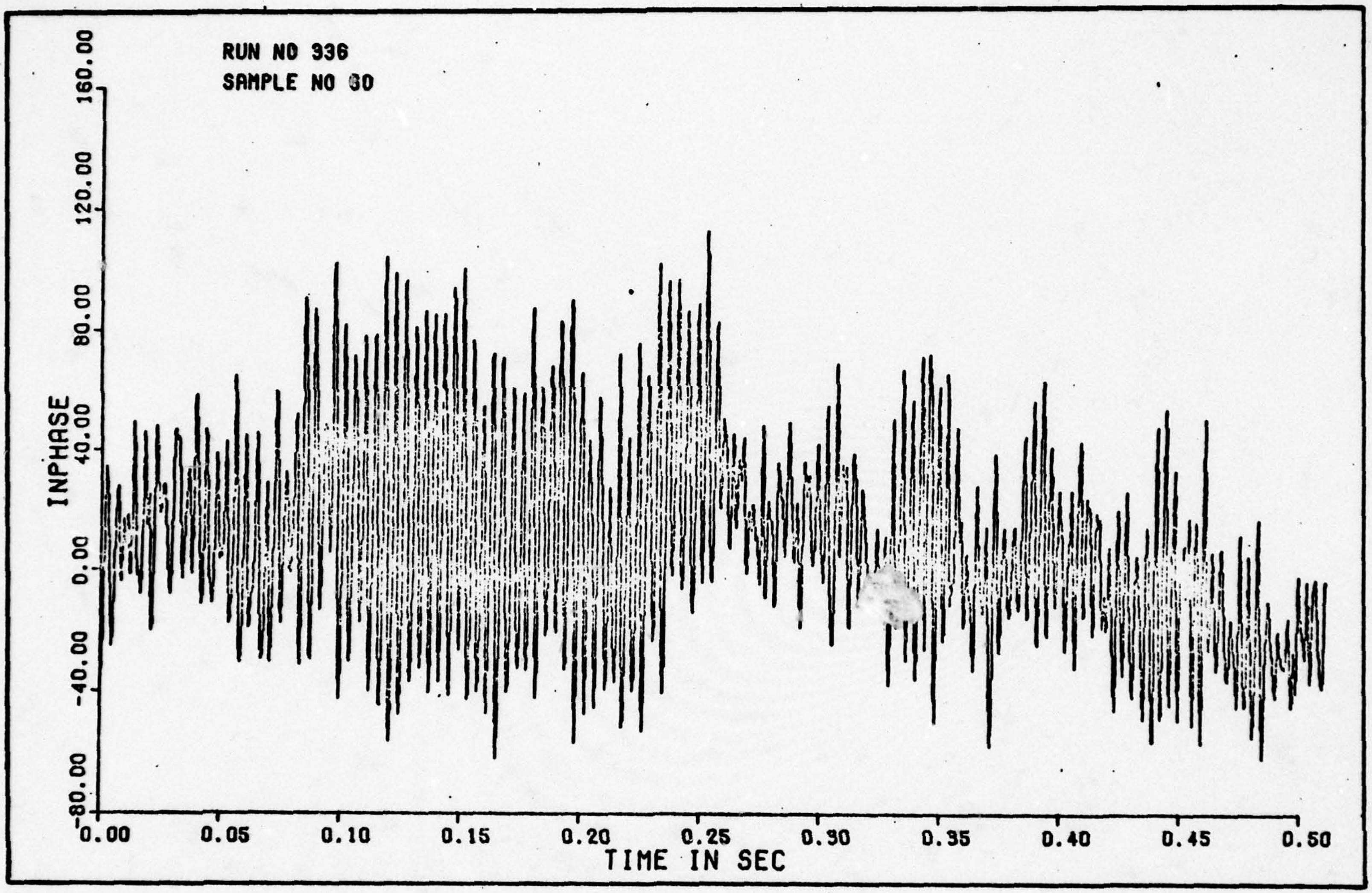


Fig. 5. Unaveraged Inphase Signal.

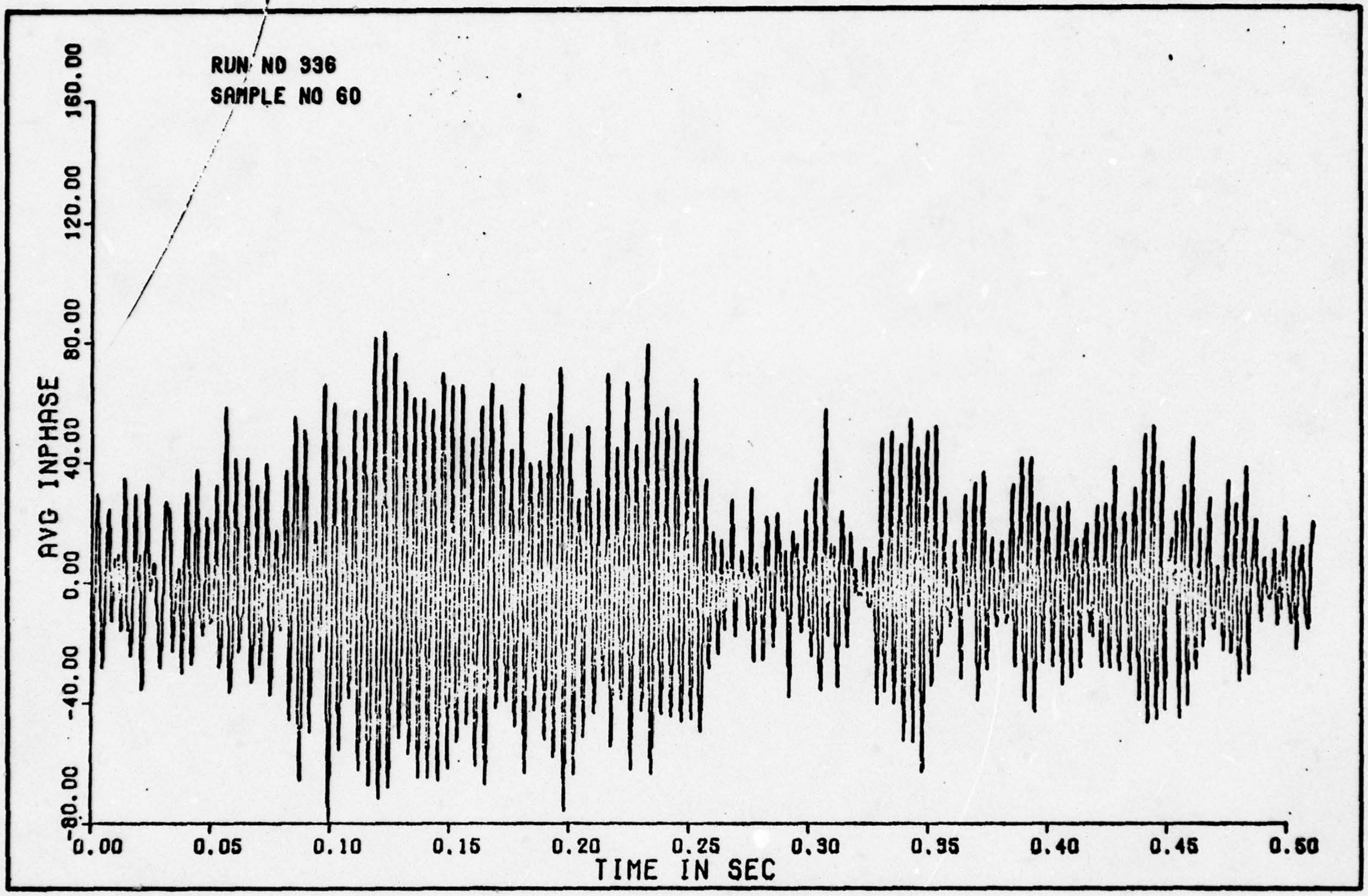


Fig. 6. Averaged Inphase Signal.

Since the averaged signals were symmetric about the zero axis, the number of periods in each sample could be determined by a zero crossing count. Since the radar returns are available in a discrete analog form, one could normalize a lower velocity (frequency) sample to a higher velocity (frequency) reference by inserting points in the unnormalized waveform at the rate T_r/T_s, where T_r is the number of periods one would observe in the reference sample and T_s is the number of periods present in the unnormalized sample. Figure 7 shows the result of normalizing a waveform upward to twice its frequency and the binary sequences which would arise after sampling and coding.

The simplest case would arise if the ratio T_r/T_s is some integer I; a normalized waveform could be obtained by retaining every Ith discrete point in the original sample, as in Fig. 7. Since the normalized waveform eventually will be infinitely clipped, the magnitudes selected for the inserted points are unimportant and only their algebraic signs are significant.

If T_r/T_s is not an integer, the points in the original sample again may be used to generate a normalized waveform; as each point is inserted into the unnormalized sample, it could be given the magnitude and sign of the nearest point already present. In this case, there is a potential for occassional errors when the inserted points are placed in the vicinity of a zero crossing in the original waveform.

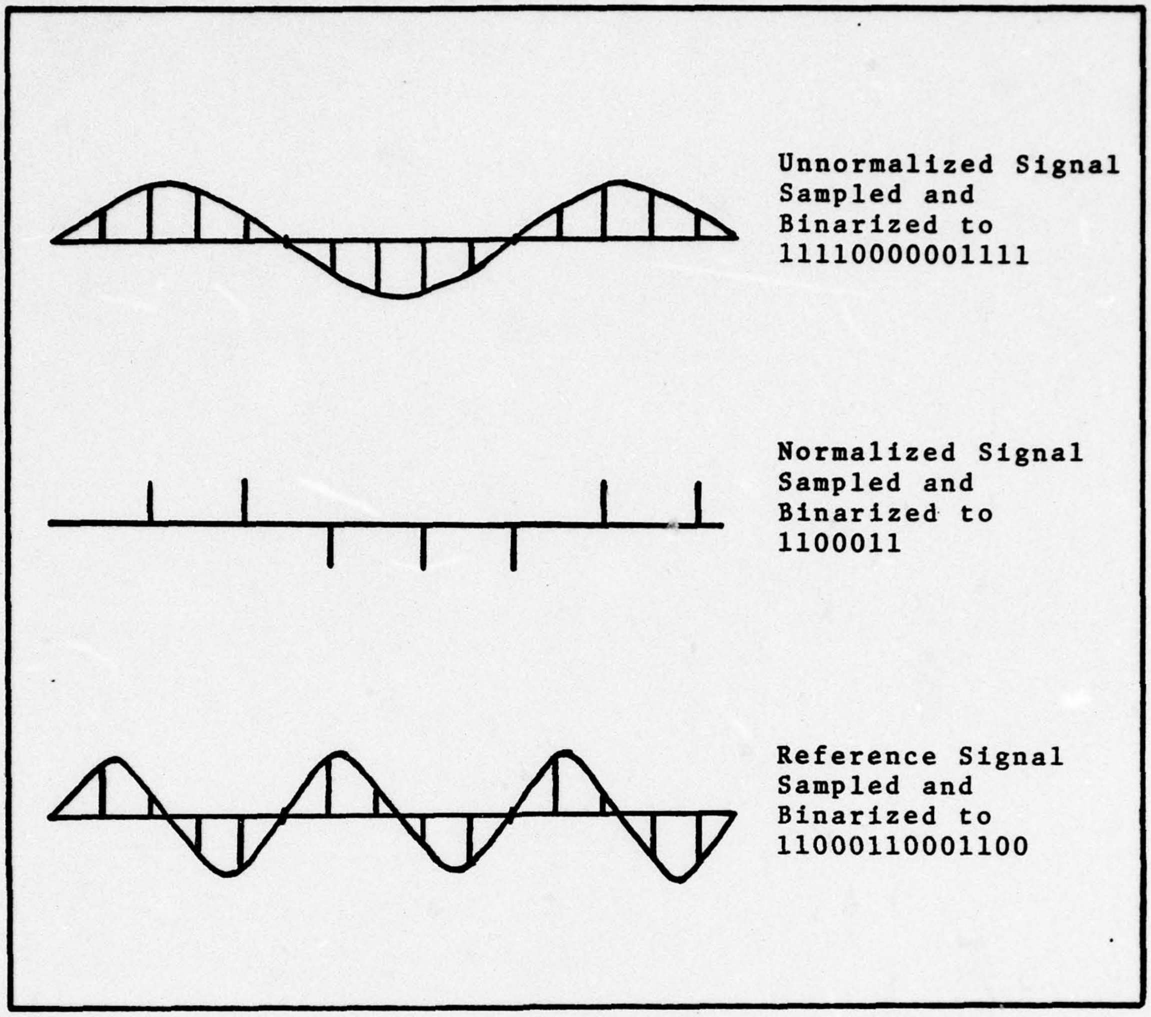


Fig. 7. Waveform Normalization.

These errors would be counted only as end effects in delayed digram frequency calculations, however, and it was thought that those calculations affected by errors might still be suitable approximations.

The clipping and coding subroutines follow the procedure described in Chapter III. At this point, the time averaged, normalized waveform is still in digitized form

and is easily converted to a binary sequence by coding all positive points as a 1 and coding all other points as a 0. The binary sequence which results, however, has somewhat fewer points than were present in the original sample. This is due to the normalization process; since the target velocities are being normalized upward to a velocity greater than that of any of the original samples, the ratio T_r/T_s is always greater than unity. As T_r/T_s is also the rate or frequency of point insertion, one would exceed the original sample length before 1024 points could be inserted. To reflect this variation in sequence length among samples, it was necessary to weight each calculated frequency of occurrence by the length of the particular sequence from which it came.

Prototype Generation

Ideally, one could generate prototypes for the FOBW method rather easily by calculating the frequencies of several hundred n-grams or delayed digrams. Ideally, at least a few of the calculated frequencies of occurrence for the members of a given class would differ substantially from those of the remaining classes. By creating the appropriate number of two class problems, one could then separate each class by the magnitude of the selected frequencies. Although prototypes were generated for each target in this fashion, it was somewhat more difficult than expected; all of the most

promising features eventually showed some dependence on target aspect angle.

The first attempt at target classification was to separate target 1 from targets 2 and 3. Targets 2 and 3 were very nearly identical and differed only by a structural modification. Target 1, on the other hand, was the only representative of its class and was particularly unique in its means of locomotion. It was also decided to begin the feature extraction process with an arbitrarily small subset of the total available data and to increase the size of the training and test sets as promising attributes emerged. This made for a reasonably short computer turn-around time in this phase of the problem and, since the attributes were selected heuristically, kept the initial amount of data to a manageable level.

The frequencies of occurrence of the delayed digrams 1-m-0, 1-m-1, and 0-m-0 were calculated for the first five runs of each target as m varied from 1 to 50. Sixty n-gram frequencies also were calculated for the same runs as n varied from 2 to 5. A training set was generated by setting aside the first sample of each run, and the selected features were tested against the remaining samples. The feature selection process was entirely heuristic, and those frequencies which appeared to have unique threshold magnitudes for each class of targets were selected as tentative prototypes. The attributes were tested individually on their ability to separate the targets; this would readily identify the set of optimal

features for consideration should separability be improved by the use of multiple attributes.

The initial training and test sets were augmented incrementally until the entire collection of data had been processed. As before, the first sample of each additional run was placed in the training set and the remaining samples were used in testing. The frequency of occurrence thresholds were adjusted to fit each new training set and were then evaluated in the cumulative test set. As the size of the training and test sets increased, the following observations were made.

1. As m increased in the 1-m-0, 1-m-1, and 0-m-0 delayed digrams, the ability of the corresponding frequency of occurrence to separate the targets did not improve over that of the features already selected. In the initial training and test set, where m varied from 1 to 50, the longest delayed digram attribute selected was 0-19-0 for the separation of targets 2 and 3, and 1-17-0 for the separation of target 1 from targets 2 and 3. When approximately half of the total available data had been examined, the process was repeated with m varying from 1 to 200 in an attempt to improve separability. None of the new attributes generated, however, could be identified as being as effective as those already in use.

2. The n-gram frequencies of occurrence were similar for all targets to such an extreme as to preclude their

use as effective attributes. While this does not necessarily infer that larger n-grams would also constitute poor features, the comparative success of the delayed digram frequencies prompted a decision to disregard the n-gram frequencies as likely attributes.

3. No single digram frequency was particularly effective when used alone. Each of the attributes selected from the initial training set eventually displayed an aspect angle dependence, and the calculated frequencies of occurrence for a given target in these situations strayed into the range of values set aside for at least one other target.

The heuristic search described above produced six delayed digram frequencies which, when used with the decision rule from the following section, best separated the targets in the final test set. The frequencies f_{1-3-0}, f_{1-5-0}, and f_{0-3-0} were given threshold values of 0.850, 0.241, and 0.092, respectively, and used to separate target 1 from targets 2 and 3; target 1 generally had 1-3-0 frequencies less than 0.850, 1-5-0 frequencies greater than 0.241, and 0-3-0 frequencies greater than 0.092, while targets 2 and 3 did not. In a similar fashion, the frequencies f_{1-3-1}, f_{1-8-1}, and f_{1-14-1} were given threshold values of 0.042, 0.064, and 0.053, respectively, and used to separate target 2 from target 3; target 2 tended to have frequencies below the threshold level for all three delayed digrams, while those of target 3 were generally higher. In selecting these

particular frequencies, consideration was given primarily to their effectiveness as a group when used with the following decision function.

Decision Function

As mentioned in the previous section, all of the selected attributes displayed some degree of aspect angle dependence. The resulting overlap of frequencies of occurrence for different targets precluded the use of a single attribute and corresponding simple decision surface for each separation. In the test of a given sample, however, if multiple features are used whose frequencies do not overlap simultaneously, one could assign a vote to the outcome of each individual test and make a class assignment by majority rule or reject the sample in the event of a tie. This decision logic is similar to that employed by committee machines, and is capable of implementing quite complex decision surfaces (Ref 5:96).

The final test set was subjected to the form of decision rule described above; each sample was first tested against the criteria for target 1 (f_{1-3-0} less than 0.850, f_{1-5-0} greater than 0.241, and f_{0-3-0} greater than 0.092), and, if at least two of these criteria were met, classified as target 1. If two or more of the criteria failed, the sample was tested against the criteria for target 2 (f_{1-3-1} less than 0.042, f_{1-8-1} less than 0.064, and f_{1-14-1} less than 0.053). If two or more of the criteria were met, the sample

was classified as target 2. Those samples which failed both the target 1 and target 2 tests were classified as target 3. Since three features were evaluated in each test, the algorithm always made a classification decision. Several attempts were made to incorporate a reject option with the use of an even number of attributes, but the added features caused more incorrect assignments to be made than were saved by rejecting.

Performance

The overall performance of the FOBW algorithm is reflected in the confusion matrices shown in Figs. 8 and 9; Fig. 8 displays the training and test set matrices for a three class problem, while Fig. 9 depicts the matrices with *targets 2 and 3 combined* in a single class. A more detailed accounting of the performance is presented in Appendices A and B, which contain the training and test set results for each run and aspect angle in the three class problem and by aspect angle alone on the two class problem.

As can be seen from Fig. 8, the algorithm achieved an overall success rate of 67.5% in the three class problem and had a reject rate of 15.5%. The low percentage of correct classification is due largely to the inability of the algorithm to distinguish between targets 2 and 3. It was mentioned earlier that targets 2 and 3 differed only by a structural modification, however, and one might suspect that they would be the most difficult to separate. Since the

True \ Assigned	1	2	3
1	26	0	3
2	2	5	4
3	4	4	14

Training Set

Target 1: 26 correct for 89.7%

Target 2: 5 correct for 45.4%

Target 3: 14 correct for 63.6%

Overall: 45 correct for 72.6%

True \ Assigned	1	2	3
1	337	3	30
2	7	98	55
3	61	158	218

Test Set

Target 1: 337 correct for 91.1%

Target 2: 98 correct for 61.2%

Target 3: 218 correct for 49.9%

Overall: 653 correct for 67.5%

177 rejects for 15.5%

Fig. 8. Training and Test Set Confusion Matrices for the Three Class Problem.

True \ Assigned	1	2/3
1	26	3
2/3	6	27

Training Set

Target 1: 26 correct for 89.7%

Target 2/3: 27 correct for 81.8%

Overall: 53 correct for 83.0%

True \ Assigned	1	2/3
1	337	33
2/3	68	529

Test Set

Target 1: 337 correct for 91.1%

Target 2/3: 529 correct for 88.6%

Overall: 866 correct for 89.6%

177 rejects for 15.5%

Fig. 9. Training and Test Set Confusion Matrices for the Two Class Problem.

similarities in these two targets are so pronounced, it would not be unreasonable to combine them into a single class and examine the performance of the FOBW algorithm in a two class problem. From Fig. 9, this results in an overall success rate of 89.6%.

The rejects shown in the confusion matrices arose from an effort to discard any sample which was recorded when the target was not established in the range gate. The majority of the runs from target 1 contained target returns which were sandwiched between empty range gate returns. The same was true but to a lesser extent for targets 2 and 3. Samples from the empty range gate typically had noise-like spectra as shown in Fig. 10, and could not be used in classification attempts.

Several plots of the inphase signal were made to determine the location and duration of the target range gate samples in the first few runs from target 1, and it was observed that samples with the target in the range gate typically had frequencies of occurrence of 0.350 or greater for the delayed digram 1-1-0. This was also evident in the data from targets 2 and 3, and it was anticipated that rejecting those samples below this 1-1-0 threshold would be a more efficient method of identifying target range gate data than plotting would be.

It became apparent, however, that this reject mechanism occassionally rejected samples corresponding to target range gate data. This was more evident in some runs than in others,

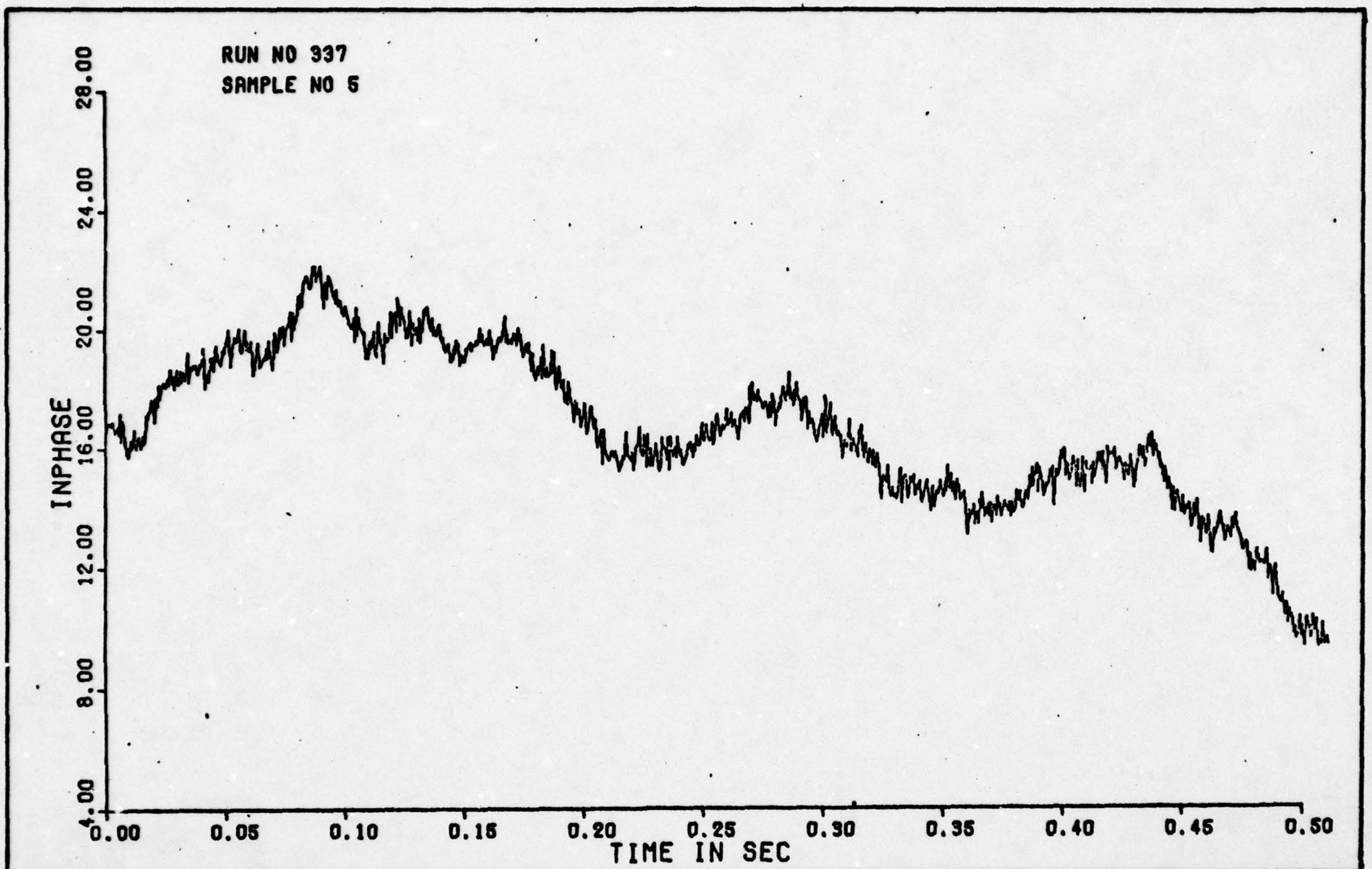


Fig. 10. Empty Range Gate Sample.

and accounts for the small number of samples in many of these runs. The number of rejects in each run was calculated by recording those rejects which occurred only when the target was established in the range gate as determined by the 1-1-0 threshold criteria.

One may also note from the appendices that those runs corresponding to aspect angles of 90 or 270 degrees were not used. The signatures of returns observed at odd multiples of 90 degrees are void of any Doppler effect; the spectral sidebands tend to collapse upon one another and thereby limit the information content of the signature. It was thought beforehand that data from these returns might consistently fail in identification attempts, and this was verified in separate trials.

V. Conclusions and Recommendations

The FOBW method has been examined as a time domain recognition algorithm for use in classifying radar signatures. From an evaluation of the performance of the method, the following conclusions were reached.

1. The FOBW method does have merit as a target classification algorithm. Considering the simplistic approaches to the time averaging and velocity normalization programs, the principle flaw in the performance of the method is the 15% reject rate. This might be improved, however, by the use of more effective averaging and normalization routines.

2. The FOBW method may be aspect angle invariant; although individual frequencies showed an angle dependence, the use of multiple frequencies did not show any correlation between errors or rejects and a particular aspect angle.

3. The higher frequency spectral components appear to contribute to target classification by this method. A short 1-m-0 digram provides an indication of the distance between the zero-crossings of a waveform, and the frequency of occurrence of two such short digrams constituted two thirds of the decision rule in the two class problem.

In light of the first conclusion reached above, the following recommendations are made.

1. The FOBW method should be applied to other data bases to determine its general applicability in target

classification problems.

2. The time averaging and velocity normalization programs should be altered to determine if the performance can be improved. Digital filtering may be a more effective method of removing the ground clutter, and it can be shown that target velocity is related to the phase of a radar return. These or other variations may result in a tighter clustering of digram frequencies and may enhance the feature selection and decision processes.

3. Variations in sample lengths should be examined. Doubling the sample length, while decreasing the total number of samples, may provide better estimates for the frequencies of occurrence. Conversely, reducing the sample length, if the frequency of occurrence estimates are still acceptable, would provide a larger test set.

Bibliography

1. Becker, P. W. *Recognition of Patterns Using the Frequencies of Occurrence of Binary Words*. Springer-Verlag, 1974.

2. Bond, F. E. and C. R. Cahn. "On Sampling the Zeros of Bandwidth Limited Signals". *IRE Transactions on Information Theory*, *IT-4*: 110-113 (September 1958).

3. Bond, F. E., C. R. Cahn, and J. C. Hancock. "A Relation Between Zero-Crossings and Fourier Coefficients for Bandwidth Limited Functions", *IRE Transactions on Information Theory*, *IT-6*: 51-52 (March 1960).

4. Davenport, W. B., Jr. and W. L. Root. *An Introduction to the Theory of Random Signals and Noise*. New York: McGraw-Hill Book Co., 1958.

5. Nilsson, N. J. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*. New York: McGraw-Hill Book Co., 1965.

6. Robinson, S. R. Reference Notes Distributed as Supplemental Lecture Materials in EE 5.27, Introduction to Fourier Optics. School of Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1976.

Appendix A

Training and Test Set Results for the Three Class Problem

Appendix A contains the training and test set results for the three class recognition problem. Entries are tabulated by target number, run number, and aspect angle, and, in the event of an incorrect classification, the target selected by the algorithm is given. For the test set results, the number of rejects in each run is also tabulated. Omitted run numbers correspond to either blank runs or data collected at 90 or 270 degrees.

Target 1 Training Set

Run No.	Aspect Angle	Classified as: Target 1	Target 2	Target 3
336	135	x		
337	315	x		
338	315			x
339	135	x		
340	225	x		
341	045			x
342	045	x		
343	225	x		
346	180	x		
347	360	x		
348	225	x		
349	22.5	x		
350	202.5	x		
351	22.5	x		
352	225	x		
353	67.5	x		
354	247.5	x		
356	67.5			x
359	292.5	x		
361	112.5	x		
362	315	x		
363	135	x		
364	337.5	x		
365	157.5	x		

Target 1 Training Set (continued)

Run No.	Aspect Angle	Classified as Target 1	Classified as Target 2	Classified as Target 3
366	337.5	x		
367	157.5	x		
368	337.5	x		
369	157.5	x		
370	315	x		
		26	0	3

Target 2 Training Set

Run No.	Aspect Angle	Classified as Target 1	Target 2	Target 3
382	157.5		x	
384	135		x	
385	315			x
386	67.5		x	
387	292.5			x
392	67.5	x		
393	247.5	x		
394	045		x	
395	22.5			x
396	22.5			x
397	202.5		x	
		2	5	4

Target 3 Training Set

Run No.	Aspect Angle	Classified as: Target 1	Target 2	Target 3
404	360			x
405	180			x
406	360	x		
407	202.5			x
408	202.5			x
409	045			x
410	045	x		
411	247.5	x		
412	247.5	x		
420	112.5			x
421	315		x	
422	135		x	
423	337.5			x
424	157.5			x
425	360			x
426	180			x
427	22.5		x	
428	202.5			x
429	045		x	
430	225			x
431	67.5			x
432	247.5			x
		4	4	14

Target 1 Test Set

Run No.	Aspect Angle	No. of Samples	No. of Samples Classified as: Target 1	Target 2	Target 3	No. of Rejects
336	135	14	13	-	1	1
337	315	20	15	1	4	4
338	315	12	12	-	-	3
339	135	12	12	-	-	8
340	225	12	12	-	-	5
341	045	17	15	-	2	4
342	045	11	11	-	-	2
343	225	3	3	-	-	3
346	180	24	21	-	3	0
347	360	26	24	-	2	1
348	225	13	13	-	-	3
349	22.5	23	20	-	3	1
350	202.5	20	19	1	-	2

Target 1 Test Set (continued)

Run No.	Aspect Angle	No. of Samples	No. of Samples Classified as Target 1	Target 2	Target 3	No. of Rejects
351	22.5	20	19	-	1	0
352	225	12	12	-	-	1
353	67.5	8	8	-	-	1
354	247.5	5	4	1	-	4
356	67.5	6	6	-	-	2
359	292.5	8	5	-	3	3
361	112.5	10	7	-	3	1
362	315	5	4	-	1	2
363	135	8	8	-	-	2
364	337.5	14	14	-	-	5
365	157.5	15	15	-	-	0
366	337.5	27	25	-	2	16

Target 1 Test Set (continued)

Run No.	Aspect Angle	No. of Samples	No. of Samples Classified as Target 1	Target 2	Target 3	No. of Rejects
367	157.5	13	11	-	2	7
368	337.5	4	3	-	1	0
369	157.5	6	4	-	2	1
370	315	2	2	-	-	0
		370	337	3	30	82

Target 2 Test Set

Run No.	Aspect Angle	No. of Samples	No. of Samples Classified as: Target 1	Target 2	Target 3	No. of Rejects
382	157.5	22	1	17	4	1
384	135	19	-	15	4	0
385	315	18	3	9	6	2
386	67.5	7	-	4	3	0
387	292.5	5	2	-	3	4
392	67.5	2	-	1	1	2
393	247.5	10	-	-	10	1
394	045	16	-	4	12	0
395	22.5	15	-	11	4	0
396	22.5	20	1	12	7	0
397	202.5	26	-	25	1	0
		160	7	98	55	10

Target 3 Test Set

Run No.	Aspect Angle	No. of Samples	No. of Samples Classified as Target 1	Target 2	Target 3	No. of Rejects
404	360	19	3	3	13	0
405	180	9	-	1	8	2
406	360	17	3	-	14	5
407	202.5	15	5	5	5	1
408	202.5	21	-	10	11	2
409	045	17	2	2	13	2
410	045	21	1	6	14	4
411	247.5	11	2	-	9	2
412	247.5	8	6	2	-	4
420	112.5	12	6	-	6	5
421	315	19	-	10	9	7
422	135	24	1	13	10	0
423	337.5	29	7	12	10	2

Target 3 Test Set (continued)

Run No.	Aspect Angle	No. of Samples	No. of Samples Classified as Target 1	Target 2	Target 3	No. of Rejects
424	157.5	25	3	17	5	0
425	360	36	-	20	16	5
426	180	35	-	20	15	4
427	22.5	31	1	17	13	5
428	202.5	30	6	10	14	3
429	045	16	1	4	11	10
430	225	21	3	6	12	3
431	67.5	11	6	-	5	5
432	247.5	10	5	-	5	14
		437	61	158	218	85

Appendix B

Training and Test Set Results for the Two Class Problem

Appendix B contains the training and test set results for the two class problem (target 1 vs. targets 2 and 3). The results for each target are presented separately and are tabulated by aspect angle alone.

Target 1 Training Set

Aspect Angle	No. of Samples Classified as Target 1	Target 2/3
360	1	-
22.5	2	-
45	1	1
67.5	1	1
112.5	1	-
135	3	-
157.5	3	-
180	1	-
202.5	1	-
225	4	-
247.5	1	-
292.5	1	-
315	3	1
337.5	3	-
	26	3

Target 2 Training Set

Aspect Angle	No. of Samples Classified as Target 1	Target 2/3
360	-	-
22.5	-	2
045	-	1
67.5	1	1
112.5	-	-
135	-	1
157.5	-	1
180	-	-
202.5	-	1
225	-	-
247.5	1	-
292.5	-	1
315	-	1
337.5	-	-
	2	9

Target 3 Training Set

Aspect Angle	No. of Samples Classified as Target 1	Target 2/3
360	1	2
22.5	-	1
45	1	2
67.5	-	1
112.5	-	1
135	-	1
157.5	-	1
180	-	2
202.5	-	3
225	-	1
247.5	2	1
292.5	-	-
315	-	1
337.5	-	1
	4	18

Target 1 Test Set

Aspect Angle	No. of Samples	No. of Samples Classified as Target 1	No. of Samples Classified as Target 2/3	No. of Rejects
360	26	24	2	1
22.5	43	39	4	1
045	28	26	2	6
67.5	14	14	-	3
112.5	10	7	3	1
135	34	33	1	11
157.5	34	30	4	8
180	24	21	3	0
202.5	20	19	1	2
225	40	40	-	12
247.5	5	4	1	4
292.5	8	5	3	3
315	39	33	6	9
337.5	45	42	3	21
	370	337	33	82

Target 2 Test Set

Aspect Angle	No. of Samples	No. of Samples Classified as		No. of Rejects
		Target 1	Target 2/3	
360	-	-	-	-
22.5	35	1	34	0
045	16	-	16	0
67.5	9	-	9	2
112.5	-	-	-	-
135	19	-	19	0
157.5	22	1	21	1
180	-	-	-	-
202.5	26	-	26	0
225	-	-	-	-
247.5	10	-	10	1
292.5	5	2	3	4
315	18	3	15	2
337.5	-	-	-	-
	160	7	153	10

Target 3 Test Set

Aspect Angle	No. of Samples	No. of Samples Classified as Target 1	No. of Samples Classified as Target 2/3	No. of Rejects
360	72	6	66	10
22.5	31	1	30	5
045	54	4	50	16
67.5	11	6	5	5
112.5	12	6	6	5
135	24	1	23	0
157.5	25	3	22	0
180	44	-	44	6
202.5	66	11	55	6
225	21	3	18	3
247.5	29	13	16	20
292.5	-	-	-	-
315	19	-	19	7
337.5	29	7	22	2
	437	61	376	85

Appendix C

FOBW FORTRAN Algorithm

Appendix C contains the FOBW FORTRAN algorithm used to classify the targets. The algorithm is commented where appropriate, and should be self explanatory.

```
      PROGRAM FOBW(TAPE7,PLOT,INPUT,OUTPUT,TAPE6=OUTPUT)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),PNF(200,2),RNRIP(1026),ARNRIP(1026)
C
C
C     PROGRAM FOBW IS THE FORTRAN ALGORITHM USED TO IMPLEMENT THE
C     FREQUENCY OF OCCURRENCE OF BINARY WORDS METHOD.  THE INPHASE
C     SIGNAL IS TIME AVERAGED, VELOCITY NORMALIZED, AND CLIPPED AND
C     CODED TO A BINARY SEQUENCE.  THIS SEQUENCE IS SCANNED FOR THE
C     FREQUENCIES OF OCCURRENCE OF THE SELECTED ATTRIBUTES, AND THESE
C     FREQUENCIES ARE THEN INPUT TO THE DECISION RULE.  PLOT ROUTINES
C     ARE ALSO AVAILABLE WHICH DISPLAY THE UNAVERAGED INPHASE SIGNAL,
C     THE TIME AVERAGED SIGNAL, THE QUADRATURE SIGNAL, AND THE PHASE
C     OF THE COMPLEX SIGNAL.
C
C     THIS PROGRAM WAS DEVELOPED FOR TARGET 3, BUT ONLY THE PRINT
C     STATEMENTS DIFFER FROM THE PROGRAMS FOR THE OTHER TWO TARGETS:
C     THE LOGIC AND TEST SEQUENCE IS THE SAME.  ADDITIONAL COMMENTS
C     ARE INCLUDED IN THE REMAINDER OF THE PROGRAM AND ARE ANNOTATED
C     BY A C AND MAY BE DISTINGUISHED FROM OPTIONAL PRINT STATEMENTS
C     ANNOTATED BY AN ASTERISK.
C
```

```
C
C     TARGET THREE
C
C
C
C
      NFILE=1
      NPP=0
    1 NREC=0
      NCOUNT=1
      NSAMP=0 $ IM1=0 $ IM2=0 $ IM3=0 $ IM4=0 $ IM5=0 $ IM6=0
    2 L=0
      M=1
    5 IF(M.GT.4) GO TO 75
   10 READ(7) NCHAN,NPOINT,(DATA(I),I=1,NPOINT)
      NREC=NREC+1
      IF(EOF(7)) 12,15
   12 PRINT*
      PRINT*
      PRINT*
      PRINT*,"              TARGET 3 VS TARGET 1"
      PRINT*
      PRINT*,"THERE ARE ",IM1," MISSES FOR ",NSAMP," SAMPLES IN FILE ",N
     #FILE
*     PRINT*,"EOF ENCOUNTERED AFTER RECORD ",NREC
      PRINT*
      PRINT*,"              TARGET 3 VS TARGET 2"
      PRINT*
      PRINT*,"THERE ARE ",IM2," MISSES FOR ",NSAMP," SAMPLES IN FILE ",N
     #FILE
*     PRINT*,"END OF FILE ",NFILE
   14 NFILE=NFILE+1
      IF(NFILE.GT.52) GO TO 500
      GO TO 1
   15 CONTINUE
```

```
C     SELECT A SAMPLE.
      IF(NFILE-26) 10,42,500
   42 IF(NFEC.LE.1) GO TO 10
      RUN=404
      IF(NPEC.GT.81) 10,65
   65 CONTINUE
      M=M+1
*     PRINT*
*     PRINT*,NPOINT," POINTS READ FROM RECORD ",NREC,", FILE ",NFILE
*     PRINT*
C     STORE THE INPHASE SIGNAL (RIP) AND QUADRATURE (Q) POINTS IN
C     SEPAPATE ARRAYS, AND CALCULATE THE PHASE (ARCTAN Q/I) OF THE
C     COMPLEX SIGNAL.
      DO 70 I=1,256
      K=L+I
      RIP(K)=DATA(2*I-1)
      Q(K)=DATA(2*I)
      IF(Q(K).EQ.0.0.AND.RIP(K).EQ.0.0) 66,67
   66 Y(K)=0.0
      GO TO 68
   67 Y(K)=ATAN2(Q(K),RIP(K))
   68 X(K)=.0005*(K-1)
   70 CONTINUE
      L=L+256
      GO TO 5
   75 CONTINUE
C     TIME AVERAVE AND NORMALIZE THE SIGNAL
      CALL AVG1
      CALL NORM(MAX,NSAMP,NFILE,IM1,IM2,IM3,IM4,IM5,IM6)
  100 CONTINUE
      GO TO 2
  500 CONTINUE
      PEWIND 7
      STOP
      END
```

```
      SUBROUTINE AVG1
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),RNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINES AVG1 AND AVG2 TIME AVERAGE THE INPHASE SIGNAL.
      DO 50 K=1,16
      IF(K.EQ.1) 20,30
   20 I=1
      M=64
      GO TO 40
   30 I=I+64
      M=M+64
   40 CALL AVG2(I,M)
   50 CONTINUE
      RETURN
      END

      SUBROUTINE AVG2(I,M)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),RNF(200,2),RNRIP(1026),ARNRIP(1026)
      SUM=0
      DO 10 K=I,M
      SUM=SUM+RIP(K)
   10 CONTINUE
      AVG=SUM/64
      DO 20 K=I,M
      ARIP(K)=RIP(K)-AVG
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NORM(MAX,NSAMP,NFILE,IM1,IM2,IM3,IM4,IM5,IM6)
      COMMON DATA(600),X(1026),Y(1026),D(1026),RIP(1026),ARIP(1026),N(10
     #24),PNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE NORM NORMALIZES THE RADIAL VELOCITY TO APPROXIMATELY
C     18 MPH.
      L=0
      L1=0
      M=0
      DO 10 K=1,1023
      IF(ARIP(K).GT.0.0.AND.ARIP(K+1).LE.0.0) L=L+1
      IF(ARIP(K).LE.0.0.AND.ARIP(K+1).GT.0.0) L=L+1
   10 CONTINUE
      PER=L/2.
      RATIO=184/PER
      DO 20 K=1,1024
      Z=RATIO*K
      INZ=Z
      IF(INZ.GT.1023) GO TO 30
      RINZ=INZ
      IF((Z-RINZ).LT.0.5) 14,16
   14 RNRIP(K)=ARIP(INZ)
      L1=L1+1
      GO TO 20
   16 RNRIP(K)=ARIP(INZ+1)
      L1=L1+1
   20 CONTINUE
   30 CONTINUE
      MAX=L1
      CALL CLIP (MAX)
C     CLIP AND CODE THE NORMALIZED INPHASE SIGNAL TO A BINARY SEQUENCE.
      CALL DGRAM(MAX,NSAMP,NFILE,IM1,IM2,IM3,IM4,IM5,IM6)
      RETURN
      END
```

```
      SURROUTINE CLIP (MAX)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),FNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SURROUTINE CLIP INFINITELY CLIPS THE INPHASE SIGNAL AND CODES
C     IT TO A BINARY SEQUENCE.
      INTEGER ONE,ZERO
      ONE=0
      ZERO=0
      DO 30 K=1,MAX
      IF(RNRIP(K).GT.0.0) 10,20
   10 N(K)=1
      ONE=ONE+1
      GO TO 30
   20 N(K)=0
      ZERO=ZERO+1
   30 CONTINUE
*     WRITE(6,40) (N(K),K=1,MAX)
*  40 FORMAT(1X,128I1//)
      RETURN
      END
```

```
      SUBROUTINE DGRAM (MAX,NSAMP,NFILE,IM1,IM2,IM3,IM4,IM5,IM6)
      COMMON DATA(600),X(1026),Y(1026),D(1026),RIP(1026),ARIP(1026),N(10
     #24),PNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE DGRAM COMPUTES THE FREQUENCIES OF OCCURRENCE OF THE
C     SELECTED ATTRIBUTES, NORMALIZES THE CALCULATED FREQUENCIES TO
C     ACCOUNT FOR VARIABLE BINARY SEQUENCE LENGTHS, AND IMPLEMENTS THE
C     DECISION RULE.
      IT1=0 $ IT2=0 $ IT3=0 $ IT4=0 $ IT5=0 $ IT6=0
      J=1
    5 M1=0
C     CALCULATE F(1-M-0).
      DO 30 K=1,MAX
      IF((MAX-K).LT.J) GO TO 40
      IF(N(K).EQ.1) 10,30
   10 IF(N(K+J).EQ.0) 20,30
   20 M1=M1+1
   30 CONTINUE
   40 CONTINUE
      IF(J.EQ.1) 41,42
   41 INR=MAX/2
      R=FLOAT(INR)
      RM1=M1
      RNM1=RM1/R
C     REJECT IF THE SAMPLE IS OUT OF THE RANGE GATE.
      IF(RNM1.LT.0.350) GO TO 875
      GO TO 45
   42 IQ=MAX/J
      INREM=MAX-J*IQ
      REM=FLOAT(INREM)
      IF((IQ/2)*2.EQ.IQ) 43,44
   43 INR=(IQ/2)*J
      R=FLOAT(INR)
      GO TO 45
```

```
  44   INR=(IQ/2)*J
       R=FLOAT(INR)+REM
  45   RM1=M1
       RNM1=RM1/R
       RNF(J,1)=RNM1
       J=J+1
       IF(J.GT.6) 50,5
  50   CONTINUE
       J=J-1
*      PRINT*
*      PRINT*,"THE VALUES OF F(1-J-0), J=1,54, ARE:"
* 40   FORMAT(1X,128I1//)
*      WRITE(6,51) (RNF(J,1),J=1,54)
* 51   FORMAT(1X,18F7.3)
       IF(RNF(3,1).GE.0.850) IT2=IT2+1
       IF(RNF(5,1).LE.0.241) IT3=IT3+1
       J=1
  55   M1=0
C      CALCULATE F(1-M-1).
       DO 80 K=1,MAX
       IF((MAX-K).LT.J) GO TO 90
       IF(N(K).EQ.1) 60,80
  60   IF(N(K+J).EQ.1) 70,80
  70   M1=M1+1
```

```
   80  CONTINUE
   90  CONTINUE
       INR=MAX-J
       R=FLOAT(INR)
   95  RM1=M1
       RNM1=RM1/R
       RNF(J,1)=RNM1
       J=J+1
       IF(J.GT.15) 96,55
   96  CONTINUE
*      PRINT*
*      PRINT*,"THE VALUES OF F(1-J-1), J=1,54,ARE:"
*      PRINT*
*      WRITE(6,98) (RNF(J,1),J=1,54)
*  98  FORMAT(1X,18F7.3)
       IF(RNF(3,1).GE.0.042) IT4=IT4+1
       IF(RNF(8,1).GE.0.064) IT5=IT5+1
       IF(RNF(14,1).GE.0.053) IT6=IT6+1
       J=1
   97  M1=0
C      CALCULATE F(0-M-0).
       DO 120 K=1,MAX
       IF((MAX-K).LT.J) GO TO 130
       IF(N(K).EQ.0) 100,120
  100  IF(N(K+J).EQ.0) 110,120
  110  M1=M1+1
  120  CONTINUE
  130  CONTINUE
       INR=MAX-J
       R=FLOAT(INR)
       RM1=M1
       RNM1=RM1/R
       RNF(J,1)=RNM1
       J=J+1
       IF(J.GT.4) 136,97
```

```
  136 CONTINUE
      J=J-1
*     PRINT*
*     PRINT*,"THE VALUFS OF F(0-J-0), J=1,54, ARE:"
*     WRITF(6,138) (RNF(J,1),J=1,54)
* 138 FORMAT(1X,18F7.3)
C     IMPLFMENT THE DECISION RULE.
      IF(RNF(3,1).LT.0.100) IT1=IT1+1
      IANS1=IT1+IT2+IT3
      IF(NSAMP.EQ.1) 600,650
  600 PRINT*
      PRINT*,"********* TRAINING SET FOLLOWS *********"
      PRINT*
      IF(IANS1.LT.2) 610,620
  610 PRINT*,"SAMPLE ",NSAMP," IN FILE ",NFILE," IS A MISS AND
     #IS MISSCLASSIFIED AS TARGET 1"
      GO TO 900
  620 CONTINUE
      IANS2=IT4+IT5+IT6
      IF(IANS2.LT.2) 630,640
  630 PRINT*,"SAMPLE ",NSAMP," IN FILE ",NFILE," IS A MISS AND
     #IS MISSCLASSIFIED AS TARGET 2"
```

```
      GO TO 900
  640 PRINT*,"SAMPLE ",NSAMP," IN FILE ",NFILE," IS A HIT"
      GO TO 900
  650 CONTINUE
      IF(IANS1.LT.2) 700,750
  700 IM1=IM1+1
      GO TO 900
  750 IANS2=IT4+IT5+IT6
      IF(IANS2.LT.2) 800,850
  800 IM2=IM2+1
      GO TO 900
  850 CONTINUE
      GO TO 900
  875 PRINT*,"REJECT"
  900 RETURN
      END
```

```
      SUBROUTINE RIPPLOT(RUN,SAMP)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),FNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE RIPPLOT PLOTS THE UNAVERAGED INPHASE SIGNAL.
      CALL PLOT(12.0,0.0,-3)
      X(1025)=0.0
      X(1026)=0.05
      CALL SCALE(RIP,6.0,1024,1)
      CALL AXIS(0.0,0.0,11HTIME IN SEC,-11,10.0,0.0,X(1025),X(1026))
      CALL AXIS(0.0,0.0,7HINPHASE,7,6.0,90.0,RIP(1025),RIP(1026))
      CALL LETTER(RUN,SAMP)
      CALL LINE(X,RIP,1024,1,0,0)
      RETURN
      END
```

```
      SUBROUTINE ARIPLOT(RUN,SAMP)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),FNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE ARIPLOT PLOTS THE AVERAGED INPHASE SIGNAL.
      CALL PLOT(12.0,0.0,-3)
      X(1025)=0.0
      X(1026)=0.05
      CALL SCALE(ARIP,6.0,1024,1)
      CALL AXIS(0.0,0.0,11HTIME IN SEC,-11,10.0,0.0,X(1025),X(1026))
      CALL AXIS(0.0,0.0,11HAVG INPHASE,11,6.0,90.0,ARIP(1025),ARIP(1026)
     #)
      CALL LETTER(RUN,SAMP)
      CALL LINE(X,ARIP,1024,1,0,0)
      RETURN
      END
```

```
      SUBROUTINE PPLOT(RUN,SAMP)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),PNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE PPLOT PLOTS THE PHASE OF THE SIGNAL.
      CALL PLOT(12.0,0.0,-3)
      X(1025)=0.0
      X(1026)=0.05
      CALL SCALE(Y,6.0,1024,1)
      CALL AXIS(0.0,0.0,11HTIME IN SEC,-11,10.0,0.0,X(1025),X(1026))
      CALL AXIS(0.0,0.0,11HPHASE (Q/I),11,6.0,90.0,Y(1025),Y(1026))
      CALL LETTER(RUN,SAMP)
      CALL LINE(X,Y,1024,1,0,0)
      RETURN
      END
```

```
      SUBROUTINE QPLOT (RUN,SAMP)
      COMMON DATA(600),X(1026),Y(1026),Q(1026),RIP(1026),ARIP(1026),N(10
     #24),NF(51,2),W(1024),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE QPLOT PLOTS THE QUADRATURE SIGNAL.
      CALL PLOT(12.0,0.0,-3)
      X(1025)=0.0
      X(1026)=0.05
      CALL SCALE(Q,6.0,1024,1)
      CALL AXIS(0.0,0.0,11HTIME IN SEC,-11,10.0,0.0,X(1025),X(1026))
      CALL AXIS(0.0,0.0,10HQUADRATURE,10,6.0,90.0,Q(1025),Q(1026))
      CALL LETTER(RUN,SAMP)
      CALL LINE(X,Q,1024,1,0,0)
      RETURN
      END
```

```
      SUBROUTINE LETTEP(RUN,SAMP)
      COMMON DATA(600),X(1C26),Y(1026),Q(1026),RIP(1026),ARIP(1025),N(10
     #24),PNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBRCUTINE LETTER LABELS THE PLOTS.
      CALL SYMBOL(1.0,6.25,0.105,6HRUN NO,0.0,5)
      CALL SYMBOL(1.0,6.0,0.105,9HSAMPLE NO,0.0,9)
      CALL NUMBER(1.735,6.25,0.105,RUN,0.0,-1)
      CALL NUMBER(2.05,6.0,0.105,SAMP,0.0,-1)
      RETURN
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      SUBROUTINE NUPLOT
      COMMON DATA(600),X(1026),Y(1026),D(1026),RIP(1026),ARIP(1026),N(10
     #24),RNF(200,2),RNRIP(1026),ARNRIP(1026)
C     SUBROUTINE NUPLOT REINITIALIZES THE BANNER EVERY FOURTH PLOT.
      CALL PLOT (12.0,0.0,-3)
      CALL DSP(7)
      RETURN
      END
```

Appendix D

Heuristic Search Procedure

Appendix D contains an example of the heuristic search procedure used to determine the delayed digram frequency of occurrence attributes and their threshold levels. It may be recalled that the first five runs of each target were set aside as an initial training set and the frequencies of occurrence of the delayed digrams 1-m-0, 1-m-1, and 0-m-0 were calculated as m varied from 1 to 50. An abbreviated version of the procedure followed in analyzing the delayed digram frequencies is shown below; the example depicts the selection of promising 1-m-0 attributes for the two class problem (target 1 vs. targets 2 and 3) as m varies from 1 to 10.

Tables 1 and 2 show the 1-m-0 delayed digram frequencies and their means for all three targets. One method of selecting attributes would be to compare the frequency means to determine if any obvious dissimilarities exist among the targets. If desired, one could also calculate the variances for an indication of how well the individual frequencies appear to cluster about each mean. The comparison of the means in Table 2 is perhaps not as striking as one would desire, and a subjective decision as to how near the means may approach one another before a particular frequency of occurrence is discarded as a likely attribute is required. Since targets 2 and 3 were considered as a single class,

the frequency of occurrence means from both targets would be considered simultaneously, also.

The heuristic selection of attributes is made somewhat easier by a graphic display of the frequency means. Figure 11 contains a plot of the frequency means for each target vs. the sampling interval. From Fig. 11, one might subjectively select the frequencies of occurrence of the delayed digrams 1-3-0, 1-5-0, 1-6-0, 1-8-0, and 1-10-0 as those attributes which display the greatest difference in magnitudes between the frequencies of target 1 and the frequencies nearest in magnitude of targets 2 or 3. It is also possible to observe the heuristic, nonexhaustive nature of such a decision; with few relatively large differences in the means, tentative proto-type selection is largely a matter of judgement.

Having selected the attributes, the threshold levels could be established so as to include as many of the target training samples as is possible; for the 1-3-0 frequency, for example, a threshold level of 0.860 would distinguish between targets 2/3 and target 1 in all but one instance (f_{1-3-0} for sample 3, target 3, is 0.791). By classifying the sample as target 1 if f_{1-3-0} is below 0.860 and classifying it as target 2 or 3 if it is greater, one could successfully classify the sample in 14 of 15 attempts.

The remainder of the attributes could be selected in a similar fashion. As mentioned earlier, as the training and test sets grow larger, one would simultaneously discard

attributes which appear less promising when applied to a larger number of samples and adjust the threshold levels of the remaining attributes to optimize their performance.

Table I

Delayed Digram 1-m-0 Frequencies of Occurrence

Target 1 1-m-0 Frequencies of Occurrence

Run No.	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	m=10
336	0.356	0.685	0.804	0.566	0.260	0.240	0.530	0.778	0.699	0.411
337	0.356	0.713	0.822	0.560	0.280	0.238	0.490	0.717	0.727	0.450
338	0.358	0.703	0.818	0.547	0.215	0.203	0.526	0.792	0.704	0.395
339	0.350	0.680	0.828	0.600	0.320	0.276	0.408	0.635	0.636	0.490
340	0.354	0.651	0.785	0.618	0.341	0.184	0.421	0.665	0.747	0.559
Mean	0.355	0.686	0.811	0.578	0.283	0.228	0.475	0.717	0.703	0.461

Target 2 1-m-0 Frequencies of Occurrence

Run No.	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	m=10
382	0.358	0.713	0.877	0.556	0.201	0.151	0.509	0.847	0.753	0.406
384	0.360	0.723	0.891	0.555	0.193	0.160	0.517	0.861	0.753	0.387
385	0.356	0.705	0.898	0.580	0.236	0.114	0.477	0.803	0.797	0.465
386	0.360	0.720	0.913	0.568	0.200	0.160	0.524	0.858	0.762	0.393
387	0.357	0.714	0.895	0.566	0.203	0.178	0.500	0.816	0.732	0.407
Mean	0.358	0.715	0.895	0.566	0.206	0.153	0.505	0.837	0.759	0.412

Table I (Continued)

<u>Delayed Digram 1-m-0 Frequencies of Occurrence</u>

Target 3 1-m-0 Frequencies of Occurrence

Run No.	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	m=10
404	0.362	0.720	0.874	0.547	0.209	0.204	0.511	0.793	0.696	0.404
405	0.352	0.703	0.865	0.575	0.260	0.193	0.477	0.739	0.727	0.455
406	0.357	0.711	0.791	0.570	0.255	0.163	0.493	0.830	0.750	0.433
407	0.358	0.715	0.861	0.559	0.201	0.173	0.513	0.838	0.737	0.400
408	0.360	0.719	0.887	0.558	0.215	0.172	0.516	0.835	0.742	0.407
Mean	0.358	0.714	0.856	0.562	0.228	0.181	0.502	0.807	0.730	0.420

Table II

__Frequency Means__

	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9	m=10
Target 1	0.355	0.686	0.811	0.578	0.283	0.228	0.475	0.717	0.703	0.461
Target 2	0.358	0.715	0.895	0.566	0.206	0.153	0.505	0.837	0.759	0.412
Target 3	0.358	0.714	0.856	0.562	0.228	0.181	0.502	0.807	0.730	0.420

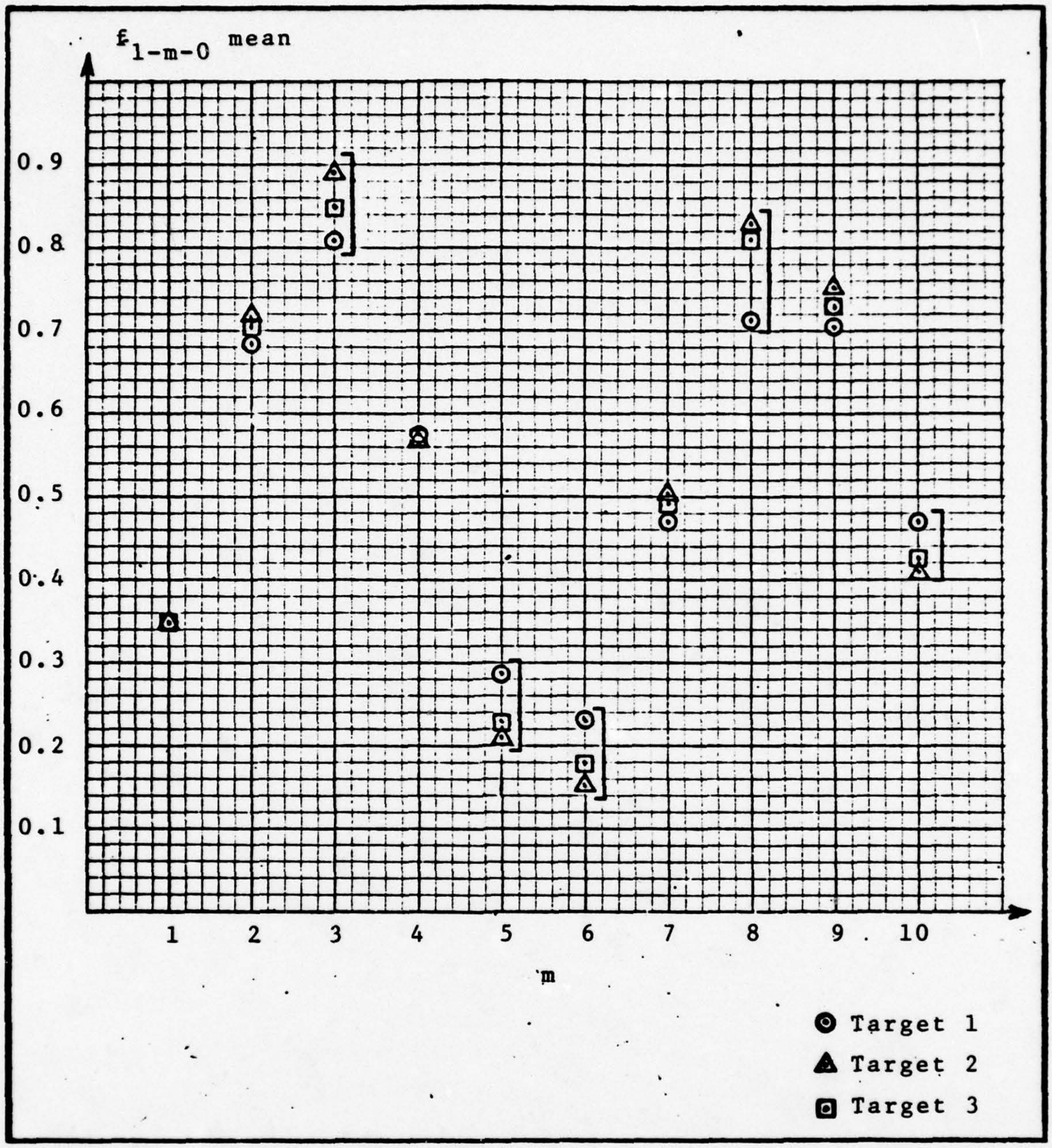


Fig. 11. m, The Sampling Interval, vs. the f_{1-m-0} Means for the Three Targets.

Vita

Delbert Glenn Kulchak was born 9 September 1939 in Coraopolis, Pennsylvania. He attended Waynesburg College, Waynesburg, Pennsylvania, and was graduated with a Bachelor of Science degree in Chemistry in 1965. He was commissioned a Second Lieutenant in the Pennsylvania Air National Guard in 1964 and attended Undergraduate Navigator Training in 1965 at James Connally AFB, Texas. Following the mobilization of his Air National Guard unit in 1968, he elected to remain on active duty with the U.S. Air Force and entered the Engineering Sciences program at the Air Force Institute of Technology in 1975.

Permanent address: 320 Moon Clinton Road
Coraopolis, Pennsylvania 15108

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (14) AFIT/GE/EE/77-25	2. GOVT ACCESSION NO. (9) Master's thesis, 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) TARGET CLASSIFICATION BY TIME DOMAIN ANALYSIS OF RADAR SIGNATURES.	5. TYPE OF REPORT & PERIOD COVERED AFIT Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Delbert [redacted] / Kulchak Glenn [redacted]	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology Wright-Patterson AFB OH 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (12) 90p.
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE (11) Dec[redacted]77
	13. NUMBER OF PAGES 80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for Public Release; IAW AFR 190-17

JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern Recognition
Pattern Classification
Target Classification
Target Identification
Radar Signature
Target Signature
Frequency of Occurrence of Binary Words
Time Domain

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this investigation is to examine a time domain pattern recognition algorithm which will identify targets by analysis of their radar signatures. The essence of the algorithm is a conversion of an analog waveform to a binary sequence, from which binary words are selected as attributes. The selection of attributes is a heuristic, nonexhaustive process, and each target is eventually described in a statistical sense by the frequency

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

012 225

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

of occurrence of the binary attributes. A relationship between this Frequency of Occurrence of Binary Words method and the Fourier transform is also shown.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)